

## Lista de Atividades - Análise de Sistemas Orientada a Objetos

### 1. Conceitos Básicos

Explique a diferença entre um sistema de informação e um software. Exemplo: Um sistema de informação de uma universidade pode incluir processos administrativos e acadêmicos, enquanto um software específico seria o sistema de matrículas.

### 2. Engenharia de Software

Descreva os principais modelos de processos de software e explique suas vantagens e desvantagens. Exemplos:

- Modelo Cascata: Fluxo linear de desenvolvimento. Vantagem: Simples e bem definido.

Desvantagem: Dificuldade em lidar com mudanças nos requisitos.

- Modelo Incremental: Desenvolvimento por partes, com entregas progressivas. Vantagem: Feedback contínuo. Desvantagem: Pode exigir maior planejamento inicial.

- Modelo Espiral: Focado em gerenciamento de riscos. Vantagem: Ideal para projetos complexos. Desvantagem: Alto custo de implementação.

### 3. Os Quatro Pilares da Orientação a Objetos

- Abstração: Representa apenas os aspectos essenciais de um objeto, omitindo detalhes desnecessários.

Exemplo:

```
java
```

```
// Classe abstrata representando um veículo
```

```
abstract class Veiculo {  
    String marca;  
    int ano;  
    abstract void acelerar();  
}
```

- Encapsulamento: Protege os dados de uma classe, permitindo acesso apenas por meio de métodos específicos.

Exemplo:

```
java
```

```
class ContaBancaria {  
    private double saldo;  
    public void depositar(double valor) {  
        saldo += valor;  
    }  
    public double getSaldo() {  
        return saldo;  
    }  
}
```

- Herança: Permite que uma classe herde atributos e métodos de outra.

Exemplo:

```
java
class Animal {
    void emitirSom() {
        System.out.println("Som genérico");
    }
}
class Cachorro extends Animal {
    void emitirSom() {
        System.out.println("Latido");
    }
}
```

- Polimorfismo: Permite que um mesmo método tenha diferentes comportamentos em classes derivadas.

Exemplo:

```
java
class Forma {
    void desenhar() {
        System.out.println("Desenhando uma forma");
    }
}
class Circulo extends Forma {
    void desenhar() {
        System.out.println("Desenhando um círculo");
    }
}
```

#### **4. Engenharia de Requisitos**

Explique a importância da Engenharia de Requisitos e cite desafios comuns no desenvolvimento de software.

Exemplo:

Um aplicativo de delivery precisa definir corretamente requisitos como tempo de entrega estimado e opções de pagamento.

## 5. Diagramas UML

Desenvolva os seguintes diagramas para um sistema de gestão de pedidos em um restaurante:

- Diagrama de Casos de Uso: Crie um diagrama mostrando os atores (Cliente, Garçon, Cozinha) e os principais casos de uso (Fazer pedido, Pagar conta, Preparar pedido).
- Diagrama de Atividades: Modele o fluxo de atividades do processo de preparação de um pedido, desde o recebimento até a entrega ao cliente.
- Diagrama de Sequência: Modele a sequência de mensagens trocadas entre Cliente, Garçon e Cozinha no processo de realização de um pedido.
- Diagrama de Classes: Crie um diagrama com as classes principais do sistema, como Pedido, Item, Cliente e Funcionário, indicando atributos, métodos e relacionamentos.

## 6. Classes e Objetos

Complete a tabela abaixo identificando elementos do código fornecido:

```
java
class Carro {
    private String modelo; // ?
    private int ano; // ?

    public Carro(String modelo, int ano) { // ?
        this.modelo = modelo;
        this.ano = ano;
    }

    public void exibirInfo() { // ?
        System.out.println("Modelo: " + modelo + ", Ano: " + ano);
    }
}
```

Código   Elemento
----- -----
`private String modelo;`   ?
`public Carro(String modelo, int ano) {`   ?
`public void exibirInfo() {`   ?

## **7. Relacionamentos entre Classes**

Explique os tipos de relacionamentos entre classes (associação, agregação, composição e herança) e forneça exemplos.

## **8. Modelagem Dinâmica**

Desenhe um diagrama de sequência para a funcionalidade de login em um sistema web.

Exemplo:

1. O usuário insere credenciais.
2. O sistema verifica a autenticidade.
3. Se válido, direciona para a página inicial.
4. Se inválido, retorna erro.

## **9. Qualidade de Software**

Quais são os atributos de qualidade de software segundo a norma ISO 25010? Como eles impactam o desenvolvimento de sistemas?

Exemplo:

- Usabilidade: Interfaces intuitivas facilitam a experiência do usuário.
- Confiabilidade: O sistema deve funcionar corretamente sem falhas frequentes.
- Eficiência: O tempo de resposta do sistema deve ser adequado.

## **10. Estudo de Caso**

Analise um sistema real (como um e-commerce ou um sistema de gerenciamento acadêmico) e descreva como a Análise Orientada a Objetos pode ser aplicada.

Exemplo:

Para um e-commerce:

- Classes: Cliente, Produto, Pedido, Pagamento.
- Casos de uso: Cadastrar produto, Adicionar ao carrinho, Finalizar compra.
- Diagramas UML: Casos de uso, Classes, Sequência.